

Are there thermodynamic cycles on the microscopic scale?

Dominik Janzing

Max Planck Institute for Intelligent Systems

Spemannstr. 38

72076 Tübingen, Germany

Email: dominik.janzing@tuebingen.mpg.de

January 6, 2017

Abstract

In a previous preprint I proposed to study the thermodynamic cost of computation and control using ‘physically universal’ cellular automata or Hamiltonians. I defined them as systems that admit the implementation of any desired transformation on a finite target region by first initializing the state of the surrounding and then letting the system evolve according to its autonomous dynamics. This way, one obtains a model of control where each region can play both roles the controller or the system to be controlled. In physically universal systems every degree of freedom is indirectly accessible by operating on the remaining degrees of freedom. In a nutshell, the thermodynamic cost of an operation is then given by the size of the region around the target region that needs to be initialized. In the meantime, physically universal CAs have been constructed by Schaeffer (in two dimensions) and Salo & Törmä (in one dimension). Here I show that in Schaeffer’s CA the cost for implementing n operations grows linear in n , while operating in a thermodynamic cycle required sublinear growth in order to ensure zero cost *per operation* in the limit $n \rightarrow \infty$. Whether this is a property of every physically universal CA has to be left to future research. After all, physical universality implies a certain kind of instability of information, which could result in lower bounds on the cost of protecting information from its noisy environment.

The technical results of the paper are sparse and quite simple. The contribution of the paper is mainly conceptual and consists in illustrating the type of thermodynamic questions raised by models of control that rely on the concept of physical universality.

1 Why thermodynamics of computation and control requires new models

The question of whether there are fundamental lower bounds on the energy consumption of computing devices has attracted the attention of researchers since the 1960s. Landauer [1] realized that logically irreversible operations like erasure of memory space necessarily require the energy $\ln 2kT$ per bit (with k denoting Boltzmann’s constant and T the temperature of the environment the information is dissipated to) due to the second law of thermodynamics.¹ Bennett [3] clarified that computation can be performed without logically irreversible operations and thus Landauer’s argument does not prove any fundamental lower bound for the energy needed by computation tasks. Ref. [4] argues that physical models of reversible computation should include the clocking mechanism (that control the implementation of logical gates) because it is not clear otherwise how to implement clocking in a thermodynamically

¹In [2] we have argued that the energy requirements for *reliable* erasure are even larger than Landauer’s bound when the state of the energy source is noisy, for instance if it is given by two thermodynamic reservoirs of different temperatures.

reversible way (after all, if both gates and clocking device are described as quantum systems then the influence of the latter on the former would, to some extent, also imply an influence of the former on the latter [5]).

It would go beyond the scope of this article to discuss the problem of physically reversible clocking of computation processes. To motivate this work, however, we should point out that the thermodynamics of clocking and synchronization is a sophisticated problem [6, 7, 8, 9]. [8], for instance, study some synchronization protocols that suggest that thermodynamically reversible synchronization requires to exchange *quantum* information, which links the a priori different tasks of reversible computation and quantum computing.²

In the same way as it is unclear whether the implementation of reversible logical operations is possible in a thermodynamically reversible way we can ask whether the implementation of unitary operations on some quantum system is possible in a thermodynamically reversible way. Regardless of how we call the physical devices controlling the implementation (we called it ‘clock’ in the case of computation processes), also the implementation of a unitary U requires to ‘change Hamiltonians’ unless U is just given by e^{-iHt} for some $t \in \mathbb{R}^+$ where H denotes the free Hamiltonian of the system under consideration. However, do we really have appropriate models for discussing the thermodynamic cost of ‘changing a system’s Hamiltonian’?

For both tasks, computation and control, we are criticizing basically the same issue: as long as the device controlling or triggering the operations (regardless of whether we call it ‘clock’ or ‘controller’) is not included in our microscopic description, we are skeptical about the claim that the operation could ‘in principle’ be implemented in a thermodynamic cycle without any energy cost.

These remarks raise the following two questions: (1) What are appropriate models for discussing resource requirements of computation and control? Given such a model, we need to ask (2) how to define *resource requirements* within the model.

To discuss the cost of ‘changing Hamiltonians’ we first recall that changing ‘effective Hamiltonians’ is what is actually done: Let the target system, for instance, be a single particle. Changing control fields actually means to change the quantum state of the physical systems surrounding the particle. In a certain mean-field limit, this state change amounts to the change of a classical field. Thus, the particles interact according to a *fixed* Hamiltonian. Taking this perspective serious, we are looking for a model where control operations are implemented by a fixed interaction Hamiltonian if the state of the surrounding quantum systems are tuned in an appropriate way.

As models for reversible computing, Hamiltonians on spin lattices have been constructed that are able to perform computation [11] by their autonomous evolution. This addresses the above criticism in the sense that these models do not require any external clocking. Instead, synchronization is achieved by the fixed and spatially homogeneous interaction Hamiltonian itself. Ref. [12, 13] goes one step further and describe Hamiltonians on spin lattices for which the result of the computation need not be read out within a certain time interval because the *time average state* encodes the result. This solves the more subtle problem that otherwise the readout required an external clock.

There are several properties that make spin lattices attractive as physical toy models of the world (and not only as model for a computing device): the discrete lattice symmetry represents spatial homogeneity of the physical laws and the constant Hamiltonian the homogeneity. By looking at lattices as discrete approximations of a field theoretical description of the physical world, even the presence and absence of matter can be seen as just being different states of the lattice. Accordingly, one can argue that spin lattices allow for a quite principled way of studying thermodynamics of computation and control because they model not only the computing device itself but also its interaction with the environment: to this end, just consider some region in the lattice as the *computing device* and the complement of that region as the *environment*.

For the purpose of addressing our thermodynamic of computing and control we propose

²Here, the formal distinction between quantum and classical clock signals as well as the conversion of time information between them is based on the rather general framework introduced in [10].

to consider spin lattices or cellular automata (as their discrete analog) that satisfy the additional condition of *physical universality* introduced in [14]. This property will be explained and motivated on an informal level in the following section.

The paper is structured as follows. Section 2 briefly motivates the notion of physical universality for both Hamiltonians and cellular automata, although we focus on the latter for sake of simplicity. Section 3 introduces the condition of physical universality formally and describes and discusses the notion of resource requirements introduced in [14], which is also the basis of this paper. Section 4 explains why CAs that are not physically universal may admit initializations of a finite region that ensure the implementation of endless repetitions of the same control operation – which admits the implementation of thermodynamic cycles in our sense. Section 5 explains why this simple construction is impossible in physically universal CAs and shows that Schaeffer’s CA does not admit sublinear growth. Whether this is a property of every physically universal CA has to be left to the future.

2 Physically universal Hamiltonians and cellular automata as possible solution

Ref. [14] introduces the notion of physical universality for three types of systems: (1) translationally invariant finite-range interaction Hamiltonians on spin lattices (2) quantum cellular automata, and (3) classical cellular automata. While (1) is the model that is closest to physics, (2) and (3) describe increasing abstractions that are useful for our purposes. Essentially, (2) is just the discrete time version of (1). We will restrict the attention to (3) because it turns out that the problem is already difficult enough for this case.

On an abstract level, the definition of physical universality coincides for all three cases: a system is called physically universal if every desired transformation on a finite target region can be implemented by first initializing the complement of that region to an appropriate state and then letting the system evolve according to its autonomous dynamics for a certain ‘waiting time’ t . For the cases (2) and (3), t is a positive integer while it is a positive real number for the case (1). Since cases (1) and (2) refer to *quantum* systems the set of possible transformations (completely positive trace preserving maps) is infinite, we should only demand that one can get *arbitrarily close* to the desired transformation via appropriate initializations and waiting times instead of being able to implement the desired transformation exactly.

Physically universal systems are intriguing because they provide a model class where every physical degree of freedom is indirectly accessible by operating on the remaining degrees of freedom in the ‘world’ and then letting the joint system evolve. In other words, the complement of the target region acts as the controller of the target region. This way, any part of the world can play both rolls controller or the system to be controlled. This is in contrast to some physical models of computation, e.g., [12], for which data and program registers are represented by different types of physical degrees of freedom. These systems are able to perform any desired transformation on the *data register* by appropriate initialization of the *program register*. The question of how to act on the program register cannot be addressed within the model. In physically universal systems, on the other hand, the preparation of *any* region can be achieved by operating on its complement. This reduces the question of how to act on some target region to the question of how to act on some ‘controller’ region around it. In turn, this controller region can be prepared by acting on some ‘meta-controller’ region around it. Although this does not solve the problem it shows at least that the boundary between controller and target region can be arbitrarily shifted. This is analog to the quantum measurement problem where the boundary between the measurement apparatus and the quantum system to be measured (the famous ‘Heisenberg cut’) can be arbitrarily shifted as long as the quantum description is considered appropriate: the transition from a pure superposition to the corresponding mixture can be explained by entanglement between the target system and its measurement apparatus. The resulting joint superposition of measurement apparatus and target system can be transferred to a mixture by entanglement with a

‘meta’ measurement apparatus and so on.

Physical universality has important thermodynamic consequences because it excludes the ability to completely protect information. This is because any system can be controlled by its surrounding, which results in information leaking into the system. This is in contrast to systems like [12] where the separation between data and program register ensures that the state of the program register remains unchanged by the autonomous evolution. Here, we don’t want to accept the latter class of models as physical models of computation because in the real world also program registers are physical systems that can be somehow accessed by actions on their environment. In other words, the information of the ‘program’ register is only safe because the model fails to describe how to act on that part of the system using the given interactions (these actions are external to the theory).

Physical universality thus gives rise to a thermodynamics in which the inability to protect information is a result of the ability to control every degree of freedom. On the one hand, the target system needs to interact with its environment otherwise we were not able to control it. On the other hand, this interaction makes entropy leaking from the surrounding into the target system. Ref. [14] defines the model class of physically universal systems for the purpose of studying this conflict on an abstract level. Here, we restrict the attention to discrete time dynamics on *classical cellular automata*. In the long run, one should certainly address our thermodynamic questions using *continuous time dynamics on quantum systems*. For a first approach to the problem, however, it is convenient to simplify the problem using classical CAs. One reason is also to make the problem more accessible to the computer science community. After all, it is one of the lessons learned from quantum information theory [15] that translating physics into computer scientific language can provide a new perspective and new paradigms. Moreover, the past two decades have shown that understanding thermodynamics via computer scientific models is also promising.³ As part of this oversimplification, we will define the thermodynamic cost of an operation simply by the size of the region in the surrounding of the target system that needs to be initialized. This will be partly justified in Section 3.2.

3 The formal setting

3.1 Notation, terminology

For the basic notation we mainly follow [19]. The cells of our CA in d dimensions are located at lattice points in $\Omega := \mathbb{Z}^d$. The *state* of each cell is given by an element of the alphabet Σ . For any subset $X \subset \Omega$, a *configuration* γ_X of X is a map $X \rightarrow \Sigma$. Let Σ^X denote the set of all configurations of X . The dynamics of the CA is given by a map $\alpha : \Sigma^\Omega \rightarrow \Sigma^\Omega$ that is local (i.e. the state of each cell is only influenced by the state of cells in a fixed neighborhood) and spatially homogeneous (i.e., it commutes with all lattice translations). Then, $\gamma' = \alpha(\gamma)$ for any $\gamma, \gamma' \in \Sigma^\Omega$, $\gamma \mapsto \gamma'$ indicates that the configuration γ evolves to γ' in one time step and $\gamma \xrightarrow{n} \gamma' = \alpha^n(\gamma)$ means that γ evolves to γ' in n time steps.

Definition 1 (implementing a function) *Let $X, Y \subset \Omega$ be finite sets and $f : \Sigma^X \rightarrow \Sigma^Y$ be an arbitrary function. Then we say a configuration $\phi \in \Sigma^{\bar{X}}$ implements f in time t if for every $x \in \Sigma^X$*

$$\phi \oplus x \xrightarrow{t} \psi_x \oplus f(x),$$

holds for some $\psi_x \in \Sigma^{\bar{Y}}$. Here, the sign \oplus denotes merging configurations of disjoint regions to a configuration of the union.

For physical universality, we also follow Schaeffer’s modified definition [19], which is equivalent to our original one:

³For instance, the principle of cooling devices [16, 17] and heat engines [18] can be illustrated using an n -bit register represented by n two-level systems or other simple discrete systems. For this model class, the relation between physics and information is most obvious.

Definition 2 (physical universality) We say a cellular automaton is physically universal if for all finite regions X, Y and all transformations $f : \Sigma^X \rightarrow \Sigma^Y$, there exists a configuration of X and $t \in \mathbb{N}$ implements f in time t .

For simplicity, we will often consider only the case $Y = X$. To explore the resource requirements of an ‘implementation’ we phrase this concept more formally in a way that is more precise about which parts of the surrounding cells matter:

Definition 3 (implementation of f) An implementation of $f : \Sigma^X \rightarrow \Sigma^Y$ is a triple (Z, ϕ_Z, t) where $Z \subset \Sigma^{\bar{X}}$ and $\phi_Z \in \Sigma^Z$ such that $\phi \oplus \phi'$ implements f in t time steps for all $\phi' \in \Sigma^{\bar{Z} \cap \bar{X}}$. Z is called the relevant region, $|Z|$ the size of the relevant region or also the size of the implementation. The range of the implementation is the side length of the smallest d -dimensional hypercube containing Z .

Note that the definition of an implementation does not imply that the relevant region has been chosen in a minimal way. Accordingly, future theorems on the resource requirements of implementations may read ‘the relevant region consists of at least ... cells.’ The range can be seen as the size of the apparatus. Assume, for instance, that Z consists of a small number n of single cells spread over a hypercube of side length $k \gg n$. Then we would still call this a ‘large’ apparatus even if n is small.

We will also study the concatenation of transformations as generalization of Definition 3:

Definition 4 (implementation of a sequence of transformations) Let X_0, \dots, X_n be finite regions and f_1, \dots, f_n with $f_j : \Sigma^{X_{j-1}} \rightarrow \Sigma^{X_j}$ be functions. An implementation of f_1, f_2, \dots, f_n is an $n+2$ -tuple $(Z, \phi_Z, t_1, \dots, t_n)$ with $t_1 < t_2 < \dots < t_n$ and $Z \cap X_j = \emptyset$ for all $j = 0, \dots, n-1$ such that $\phi_Z \oplus \phi'$ implements f_j in t_j time steps for all $\phi' \in \Sigma^{\bar{Z} \cap \bar{X}_{j-1}}$. The size of the implementation is $|Z|$ and the range of the implementation is the side length of the smallest d -dimensional hypercube containing Z .

Consider, for instance, a CA with binary alphabet $\Sigma = \{0, 1\}$. Assume the task is to implement a not gate on the same bit n times on some target bit. Then the desired functions read $f_1 = \text{NOT}, f_2 = \text{ID}, f_3 = \text{NOT}, \dots$ and the numbers t_j specify the time instants for which the autonomous dynamics has implemented another NOT gate on our target bit, given that some region Z has been initialized to the state ϕ_Z . Note that this definition of an implementation of a sequence does not cover the case where the target regions X_0, \dots, X_{n-1} are accessed by an external operation during the time where the system evolves according to its autonomous dynamics. This yielded a stronger notion of an implementation that we don’t consider here although it would also be interesting.

3.2 Defining ‘thermodynamic cost’ of operations

Here we will consider the size of the relevant region as the thermodynamic cost of an implementation. This first approximation is justified by the following idea: a priori, the state of each cell is unknown, i.e., we assume uniform distribution over Σ . According to Landauer’s principle it then requires the energy $kT \ln |\Sigma|$ to initialize one cell to the desired state. This way, the thermodynamic cost of the initialization process is simply proportional to the number of cells to be initialized. This view will be further discussed at the end of this subsection.

We are now able to phrase the following thermodynamic questions using the language above:

- **Zero cost per implementation:** Let $f : \Sigma^X \rightarrow \Sigma^X$ be an arbitrary function and $(Z_n, \phi_{Z_n}, t_1, t_2, \dots, t_n)$ be implementations of the powers f, f^2, f^3, \dots, f^n . Is it possible that $|\phi_{Z_n}|/n \rightarrow 0$?
- **Zero cost of information storage per time:** For each $n \in \mathbb{N}$, let (Z_n, ϕ_{Z_n}, n) be the implementation of the identity on region $X \mathbb{Z}^d$, that is, the configuration ϕ_n ensures that the information contained in the region X is restored after n time steps. Is it possible to achieve that $|\phi_{Z_n}|/n \rightarrow 0$?

We will show that both questions have a negative answer in Schaeffer’s CA [19], while there exist CAs that are not physically universal for which both answers are positive. We leave it as an open question whether non-zero cost are implied by physical universality. However, we give some intuitive arguments that suggest that physical universality makes it at least more difficult to achieve zero implementation cost per operation or zero cost for information storage per time.

It is certainly an oversimplification to identify the size of the region that needs to be initialized with the thermodynamic cost of an implementation. Consider, for instance, a physical many particle system where each cell is a physical system that is weakly interacting with its neighbours. This ensures that the total energy of the composed system is approximately given by the sum of the individual systems. Assume, furthermore, that the state $0 \in \Sigma$ corresponds to the ground state, that is, the state of lowest energy. In the limit of low temperatures, this state has probability close to 1, which implies that initializing the lattice to the all-zero state does not require significant free energy resources. In this case, however, it requires significant free energy resources to set a cell to *any state other than* 0 and the resource requirement then depends on the number of cells that need to be in a non-zero state (which may correspond to the number of particles in physics).

On the other hand, identifying the number of cells to be initialized with the thermodynamic cost, can also be justified from the following point of view: assume we are not interested in the amount of free energy that is required for one specific transformation. Instead, we only ask whether the amount increases sublinearly or not. Assuming, in the above physical picture, non-zero temperature (although it may still be low, which favors the state 0), initializing n states to 0 *with certainty* yet requires an amount of free energy *of the order* n . This way, the asymptotic behaviour of resource requirements is unaffected by the details of the physical hardware assumptions.

4 Cost per operation in Turing complete CAs

To take a simple toy example, we consider the control task of repeatedly turning a target bit on and off and never stop. Intuitively, this process already reminds us of an endless program, i.e., a program with an endless loop. Think, for instance of the following pseudo code:

```
while 1  $\neq$  1 do
   $a := 1$  ;
   $a := 0$ ;
end while
```

Every Turing complete CA is, by definition of Turing completeness, able to implement this program. Here, we call a CA Turing complete if there is a finite configuration (i.e., a configuration with finitely many non-zeros, given we have previously given one element of Σ the name ‘0’) that simulates a universal Turing machine when ‘halting’ is defined as one previously selected cell being non-zero. Note, however, that *finite configuration* does not imply *finite resources* in our sense. ‘Finite configuration’ means that all but a finite number of cells are in the zero state while ‘finite resources’ means that all but a finite number of cells are in an *unknown* state. If a simulation of a Turing machine by a CA *requires* all but a finite number of cells to be zero because non-zero far-distant cells would perturb the simulation if they are not zero, this amounts to infinite resources in our sense. In this case, it amounts to a non-trivial question how the size of the relevant region grows with the number of loops one wants to implement. However, as long as we do not demand physical universality, we can easily modify Turing complete CAs in way that they are able to implement an infinite loop with finite resources, as we will explain now.

4.1 Conway’s Game of Life

We first consider the implementation of our target operation ‘infinite bit switch’ in a well-known cellular automaton, namely Conway’s Game of Life. It is a CA in two dimensions,

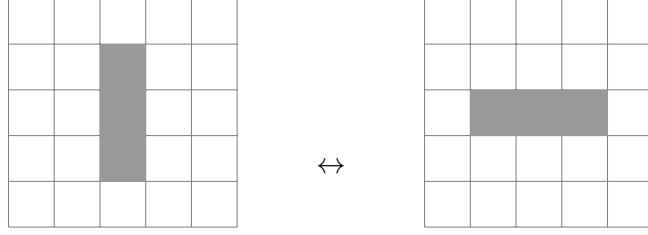


Figure 1: A simple configuration in Conway's Game of Life that yields a dynamical behaviour with period 2. The system changes between the two configurations on the left and the right hand side, respectively. 'Alive' and 'dead' cells are indicated by grey and white, respectively.

each cell being 'alive' or 'dead', i.e., formally each cell is just one bit. The rules are [20]:

- (1) Any live cell with fewer than two live neighbours dies, as if caused by under-population.
- (2) Any live cell with two or three live neighbours lives on to the next generation.
- (3) Any live cell with more than three live neighbours dies, as if by over-population.
- (4) Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

To implement the bit flip, as desired, we find simple oscillating patterns in [20]: The 'Blinker' has period 2, as shown in Figure 1. We now focus on the space requirements of this 2-cycle and recall that space requirements in our sense refer to the amount of space that needs to be initialized to a specific value. For the Blinker to work, it is essential that there are no 'particles' in the direct neighbourhood that disturb the patterns. Whenever there is a region outside which the state is not known at all, this complementary region contains with some probability a pattern that moves towards the blinker and disturbs its cycle. It is therefore possible, that, without having some control about the entire space, we cannot guarantee that the blinker works forever.

4.2 Modified Game of Life with impenetrable walls

There is, however, a simple modification of the Game of Life for which we can ensure that the blinker works forever although we only control the state of a finite region. To this end, we augment each cell by an additional third state 'brick' ■, indicated by black color, that blocks the diffusion from the surrounding. The transition rule of the new CA now consist of the following rules:

- (0) a cell being in the state ■ remains there forever. (1)-(4) as before, with the convention that the brick ■ counts as □ for its neighbors.

The idea of bricks is that they can form a 'wall' around our blinker that protects it from the influence of its surrounding (which can be in an unknown state). In physical terms, the wall protects the blinker from the heat of the environment, as shown in Figure 2.

4.3 Reversible CA: Margolus' billard ball model

To get one step closer to physics and account for the bijectivity of microscopic dynamics in the physical world, we now consider reversible CAs, i.e., CAs in which every state has a unique predecessor, which is not the case for Game of Life. We now show that even reversible CAs exist that admit perfect protection of an implementation of an infinite loop, which results in zero cost per operation.

Margolus' billard ball model CA [21] is a CA in 2 dimensions whose update rules are defined on Margolus neighbourhoods, i.e., there are two partitions of the grid into blocks of 2×2 cells describing the updates at even and odd time instants: At even time instances, the update is done on the blocks $\{(2i, 2j), (2i, 2j + 1), (2i + 1, 2j), (2i + 1, 2j + 1)\}$, at odd times it is done on the blocks $\{(2i - 1, 2j - 1), (2i - 1, 2j), (2i, 2j - 1), (2i, 2j)\}$, as visualized

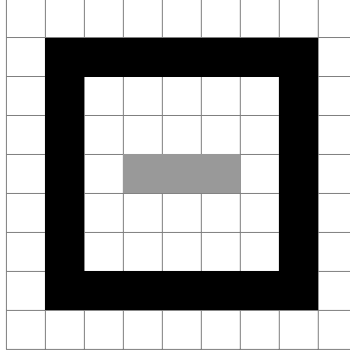


Figure 2: The blinker surrounded by a wall of ‘bricks’, which protect it from uncontrolled perturbations from its environment.

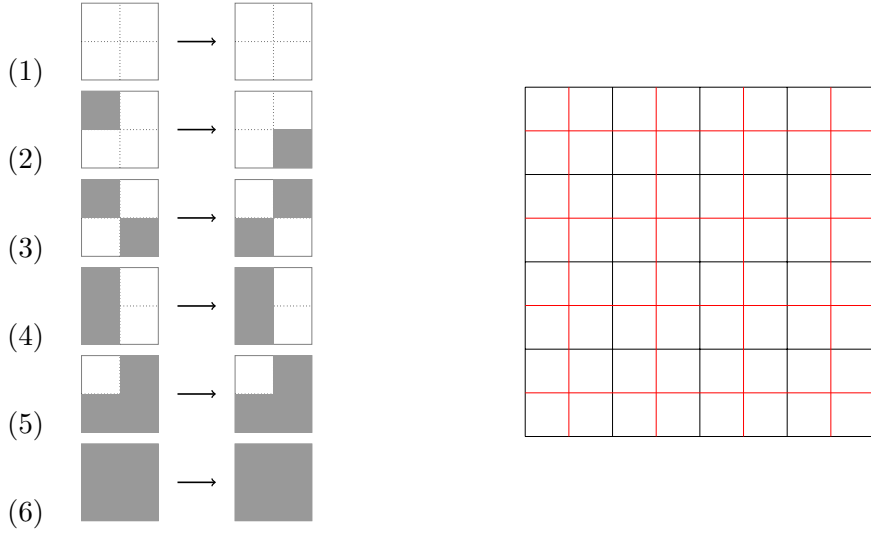


Figure 3: Left: Transition rules of Margolus’ billiard ball model CA. Right: the two different partitions are indicated by the black and the red grid.

by the black and the red grid in Figure 3, right. For each such block, the update rules are shown in Figure 3, left.

As noted in [19], the billiard ball CA is not physically universal since it allows for impenetrable walls [21]. We will use such walls to implement a bit switching process that continues forever although only a finite region has been initialized. A simple example is shown in Figure 4. In the sense of the present paper, this CA implements the NOT operation in a thermodynamic cycle since there are no resource requirements per operation because there is not need to initialize the cells outside the wall.

5 Resource requirements in physically universal CAs

5.1 Schaeffer’s physically universal CA

Schaeffer [19], see also [22], constructed a physically universal CA that is close to Margolus’ billiard ball model CA. The update rules are shown in Figure 5. We now discuss a rather primitive solution of implementing our bit switching task in Schaeffer’s CA. Its resource requirements grows at least linearly in n , which seems rather bad. Yet, we will later show that linear growth is optimal. We first observe that the CA admits free particle propagation in diagonal direction, a fact that is heavily used in the proof for physical universality [19].

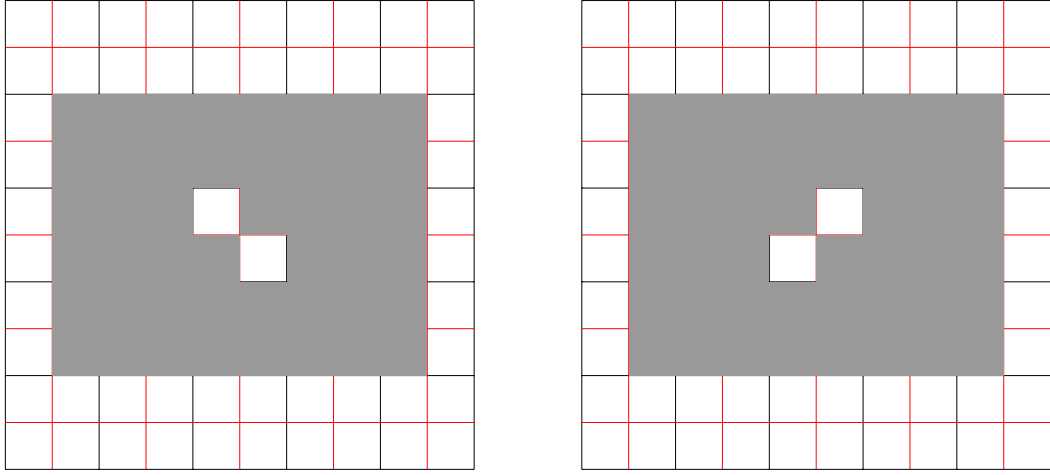


Figure 4: Configuration of BBMCA that implements bit switching forever: applying Rule (3) to the black partitioning takes the configuration on the left hand side to the one on the right hand side. Then, an update according to the red partitioning leaves the state unchanged due to Rule (5). Applying Rule (3) to the black partitioning takes the configuration on the right hand side back to the one on the left hand side. Again, updating according to the red partitioning has no effect.

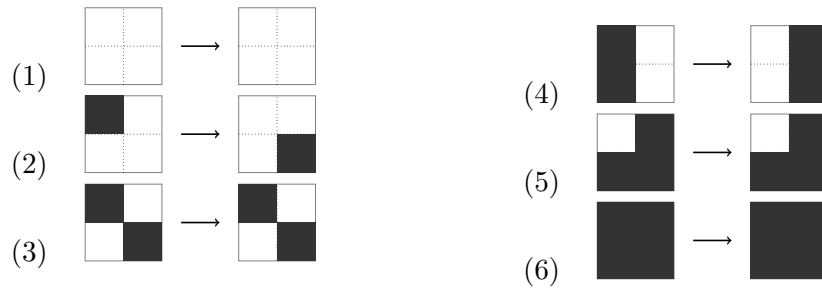


Figure 5: Transition rules of Schaeffer's physically universal CA.

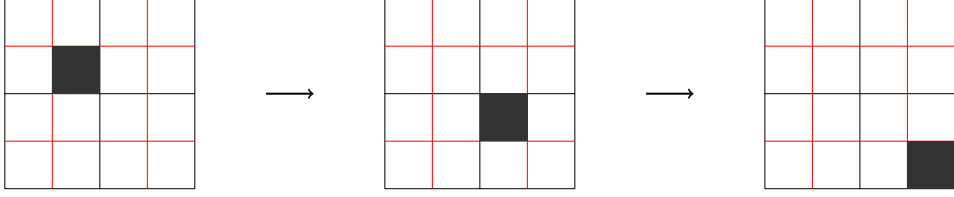


Figure 6: Free particle propagation in Schaeffer’s physically universal CA: the configuration on the left turns into the one in the middle by applying Rule (2) to the red partitioning. The middle configuration turns into the right one by applying the same rule to the black partitioning.

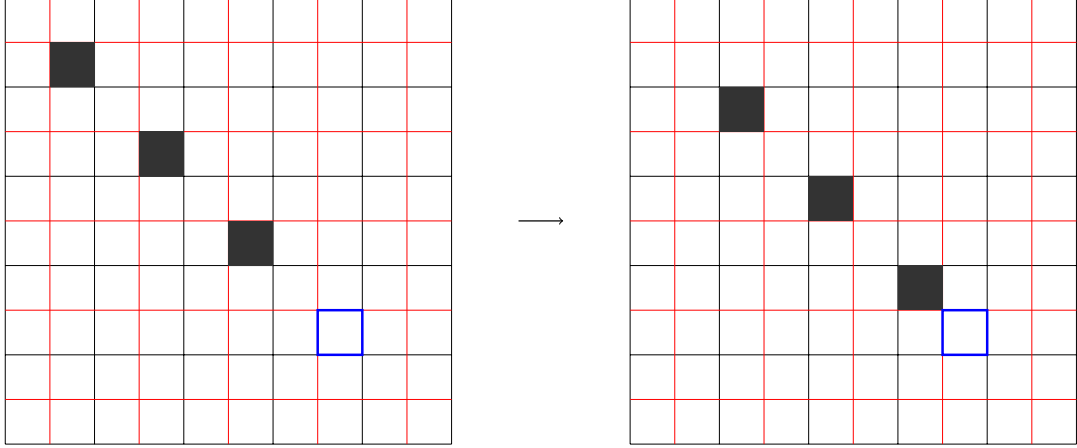


Figure 7: Beam of 3 propagating particles which implement the turning on and off of the blue target cell 3 times.

Figure 6 visualizes this motion. We now use a ‘beam of particles’ in diagonal direction in which a particle and a whole alternate, as shown in Figure 7. Then choose a target bit along the diagonal, as indicated by the blue square in Figure 7. Just by waiting, this bit is turned on and off when particles and wholes appear, respectively. The resource requirements of this implementation are large: not only does it require to correctly locate particles and wholes, it also requires to keep the space around the beam empty to protect the beam from collisions.

Apart from being costly from the thermodynamic point of view, the implementation is also ‘not nice’ in other respects: compared to the simplicity of our control problem, the initialization is rather complex. Assume, for comparison, the following general control task: given some arbitrary binary string b of length $2n$, the target bit is supposed to attain the value b_j at time j . Then, the above beam solves this task for the special case where $b = 101010 \dots 10$. The general task can obviously be solved by the same procedure as above: just locate particles and wholes according to b . The fact that the solution of the simple special case is based on the same principle suggests that it is a ‘bad’ solution; it is inappropriately complex compared to the simplicity of the task. In a way, it reduces a simple control operation to one that seems more complex. This raises the question of what one wants to call a *solution of a control task*.

To come back to the thermodynamic question, one may wonder whether there are smarter implementations of the bit switch process where the resource requirements do not grow linearly in n . We can easily show that the *range* of the implementation of the n -fold bit switch grows linearly in n . To this end, we first need the Diffusion Theorem of [19]:

Theorem 1 (diffusion) *Let X be a $2n \times 2n$ square and ϕ be a configuration such \bar{X} is empty. Then X is empty after $2n$ steps.*

For a square of odd sidelength k , one can conclude that it is empty after $k + 1$ steps. We

then find:

Theorem 2 (range of restoring a bit after t timesteps) *Let (Z, ϕ_Z) such that ϕ_Z implements the identity on site $(0, 0)$ after time t . Then the range of Z is at least $t - 1$.*

Proof: Clearly, sites that are further away than t matter only after more than t time steps because signals propagate by at most one site per step. Assume, on the other hand, Z were contained in a square of sidelength $s < t - 1$. Let ϕ be a configuration whose restriction to Z coincides with ϕ_Z and contains only zeros outside Z . Then the diffusion lemma states that ϕ develops into a state ϕ' whose restriction to Z is empty for all $t' \geq s + 1$ in contradiction to the assumption that ψ implements the identity after t steps. \square

Let us state a few remarks on similar results for the one-dimensional physically universal CA in [23]. It is also based on the intuition of propagating and interacting particles and uses particles that move with two different speeds, namely 1 or 2 sites per time step to the left or the right. One can easily show that also for this CA the range of an implementation restoring a bit after the time t grows linearly in t . This is because Lemma 2 in [23] is also a diffusion theorem similar to Theorem 1. It states (among other things) the following: whenever all particles are contained in the region $\{0, 1, \dots, n - 1\}$, one observes free particle propagation after t_n steps, where t_n grows only with $O(n)$. Moreover, the particles are then contained in the interval $[-2t_n, n + 2t_n]$. Assume now that n is even and that some configuration in $[0, n/2 - 1] \cup [n/2 + 1, n]$ implements bit restorage on cell $n/2$. Since all the particles propagate by either of the velocities $-2, -1, 1, 2$, the cell $n/2$ is empty after $2t_n + n/2 \in O(n)$ steps. If n grows sublinearly in t , the bit cannot appear in cell $n/2$ after t steps.

Moreover, we have:

Theorem 3 (range of implementing n transformations) *Given a CA for which the Diffusion Theorem 1 holds. For some region $X \subset \mathbb{Z}^d$ let $f : \Sigma^X \rightarrow \Sigma^X$ be arbitrary. Let $(Z, \phi_Z, t_1, t_2, \dots, t_n)$ with t_1, t_2, \dots, t_n different, be the implementation of f, f^2, \dots, f^n . Then its range is at least $n - 1$, that is, Z is not contained in a square of sidelength smaller than $n - 1$.*

The proof is the same as the one of Theorem 2. Obviously, $t_n \geq n$. To ensure that X is not empty after n steps we need to initialize a square of size $(n - 1) \times (n - 1)$ at least.

Given our view as interpreting the relevant region as the technical apparatus implementing the transformation, we can rephrase the Theorem by saying that the apparatus does not fit into any square of size smaller than $(n - 1) \times (n - 1)$.

Since the range is a rather coarse measure for the resource requirements (which is just convenient for proofs) we now study the size of the relevant region required for a rather elementary control task, namely to restore a bit n times:

Theorem 4 (cost for restoring a bit n times) *Let ID denote the identity on one target cell and $(Z, \phi_Z, t_1, t_2, \dots, t_n)$ be an implementation of ID, ID, \dots, ID in the CA in [19]. Then Z contains at least $n/4$ cells.*

Proof: First extend Z to a region $Z' \supset Z$ such that Z' consists only of complete 2×2 block with respect to a fixed partitioning. Z' may contain also the target cell, but this is irrelevant for the argument below.

We now make heavily use of the techniques developed in the proof of Theorem 4 in [19]: we also define an ‘abstract’ CA that consists of three states $\{0, 1, \top\}$, where \top denotes a ‘wild card’ that stands for an uncertain state. The purpose of the abstract CA is merely to keep track of how uncertain states propagate in the concrete CA. Ref. [19] describes a pretty simple set of update rules for the abstract CA, whose details are not needed. The essential observation that we need is that \top particles show free particle propagation as long as the ‘forbidden’ configurations



and their rotated versions do not occur. Here, the grey boxes indicate \top state, i.e., cells that are either white or black.

We now assign the value \top to all cells in Z' , representing the fact that we don't know their truth values because we don't know how ϕ_Z looks like. Then we assign the value 0 (i.e. white) to all cells in the complement of Z' . This is possible because, by assumption, cells outside Z' do not matter. This way, the forbidden configurations are avoided and the dynamics of the abstract CA can be described by free propagation of the \top -particles. Then each of the \top particles moves in either of the directions North-East, South-East, South-West, North-West. Let $|Z'| = k$, i.e., we have a free propagation of k \top -particles. Then an arbitrary given target cell can attain a non-zero value at most k times. Assume the target cell started in the state 1 (i.e. black). Implementing ID n – times requires that it attains the state 1 in n time steps. Hence, $k \geq n$. Since $k \leq 4|Z|$ by construction of Z' we conclude $4|Z| \leq n$. \square

Theorem 4 can easily be applied to our task of n -fold NOT since the latter amounts to implementing the identity for all t_j with even j .

It is unclear whether some of these insights apply to a general physically universal CA. The question whether there exists a physically universal CAs that do not satisfy the Diffusion Theorem has already raised by Schaeffer [19], which seems related to our thermodynamic questions since diffusion is what makes information so terribly unstable.

It is, however, clear that in any physically universal CA a configuration of a finite region is unstable in the following sense:

Theorem 5 (instability of patterns) *For some physically universal CA, let $Z \subset \mathbb{Z}^d$ be a finite region that is initialized to the state ϕ_Z . Let the states of all cells outside Z be unknown with some probability distribution P that assigns non-zero probability to every possible state in $\Sigma^{\mathbb{Z}^d \setminus Z}$. Then, for any configuration ϕ'_Z of Z there is a t such that ϕ_Z evolves to ϕ'_Z with non-zero probability.*

Proof: Choose a function $f : \Sigma^Z \rightarrow \Sigma^Z$ with $f(\phi_Z) = \phi'_Z$. By physical universality, there is a configuration of the complement of Z implementing f for some t . Since only the restriction of the configuration to a finite region matter (cells that are further away than t sites do not matter) the set of all configurations implementing f have non-zero probability. \square

The non-existence of impenetrable walls is only the most intuitive consequence of this observation. Further, less obvious consequences remain to be discovered.

6 Conclusions

Common discussions on thermodynamic irreversibility of operations often focus on entropy generation while they substantially differ respect to the underlying notion of entropy (e.g. Boltzmann entropy, Shannon respective von Neumann entropy, or Kolmogorov complexity [24, 25, 26]). This may be due coarse graining [27], or because complexity also contributes to physical entropy by definition [24, 25], or because entropy leaks into the system from its environment.

Irreversibility in physically universal CAs or Hamiltonian systems is not due to entropy production – at least not in any obvious sense. Instead, every evolution is to some extent irreversible simply because one has no access to the evolution, it just continues forever. Simulating the inverse evolution on some target system involves sophisticated initialization of a large number of cells in the surrounding (acting as the controller). Since this initialization is typically destroyed by the autonomous evolution of the system, restoring state of the joint system of target *and* its controller involves a sophisticated initialization of a ‘meta-controller’, which, in turn, will then be destroyed by the evolution. The question of how to reverse the dynamics of one system without disturbing the state of its surrounding thus raises the same question for an even larger system.

The idea that control operations, even when they are unitary, imply heat generation in the controlling device, is certainly not new. However, physically universal CAs and Hamil-

tonians may allow us to look at the idea from a new perspective because they admit to describe target, controller, meta-controller and so on, in a unified way since all of them are just regions of cells. Moreover, physically universal CAs formalize the conflict between controllability and isolability of a system in a principled way. This is because physical universality, which formalizes the ability to control subsystems, implies instability of information, although quantitative results have to be left to the future. Here we have shown that in the existing constructions of physically universal cellular automata information is ‘terribly’ unstable – for instance, in the sense that the resource required for protecting information grows linearly in time.

The intention of this article is to inspire other researchers to explore implications of physical universality rather than exploring properties of specific constructions of CAs. Here we have discussed properties of Schaeffer’s construction only to illustrate how to work with our notion of resource requirements in the context of a physically universal CA.

Acknowledgements: I would like to thank Scott Aaronson and Luke Schaeffer for interesting discussions on related questions.

References

- [1] R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.*, 5:183–191, 1961.
- [2] D. Janzing, P. Wocjan, R. Zeier, R. Geiss, and Th. Beth. Thermodynamic cost of reliability and low temperatures : Tightening Landauer’s principle and the Second Law. *Int. Jour. Theor. Phys.*, 39(12):2217–2753, 2000.
- [3] C. H. Bennett. Logical reversibility of computation. *IBM J. Res. Develop.*, 17:525–532, 1973.
- [4] D. Janzing and Th. Beth. Are there quantum bounds on the recyclability of clock signals in low power computers? In *Proceedings of the DFG-Kolloquium VIVA*, Chemnitz, 2002. See also preprint [arXiv:quant-ph/0202059](#).
- [5] D. Janzing and T. Decker. How much is a quantum controller controlled by the controlled system? *Applicable Algebra in Engineering, Communication and Computing*, 19(3):241–258, 2008.
- [6] D. Janzing and B. Steudel. Quantum broadcasting problem in classical low power signal processing. *Phys. Rev.*, A(75):022309, 2007.
- [7] D. Janzing and T. Beth. Bounds on the entropy generated when timing information is extracted from microscopic systems. preprint [arXiv:quant-ph/0301125](#), 2003.
- [8] D. Janzing and T. Beth. Synchronizing quantum clocks with classical one-way communication: Bounds on the generated entropy. preprint [arXiv:quant-ph/0306023v1](#), 2003.
- [9] P. Erker, M. Mitchison, R. Silva, M. Wood, Brunner M., and M. Huber. Autonomous quantum clocks: how thermodynamics limits our ability to measure time. preprint [arXiv:1609.06704](#), 2016.
- [10] D. Janzing and T. Beth. Quasi-order of clocks and their synchronism and quantum bounds for copying timing information. *IEEE Transactions Information Theory*, 49(1):230–240, 2003.
- [11] N. Margolus. Parallel quantum computation. In W. Zurek, editor, *Complexity, Entropy, and the Physics of Information*. Addison Wesley Longman, 1990.
- [12] D. Janzing and P. Wocjan. Ergodic quantum computing. *Quant. Inf. Process.*, 4(2):129–158, 2005.
- [13] D. Janzing. Spin-1/2 particles moving on a 2D lattice with nearest-neighbor interactions can realize an autonomous quantum computer. *Phys. Rev. A*, 75:012307, 2007.

- [14] D. Janzing. Is there a physically universal cellular automaton or Hamiltonian? preprint [arXiv:1009.1720](#).
- [15] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [16] J. Fernandez, S. Lloyd, T. Mor, and V. Roychowdhury. Algorithmic cooling of spins: A practicable method for increasing polarization. *Int. Journ. Quant. Inf.*, 2(4):461–467, 2004.
- [17] A. Allahverdyan, K. Hovhannisyan, D. Janzing, and G. Mahler. Thermodynamic limits of dynamic cooling. *Phys. Rev. E*, E(48):041109, 2011.
- [18] D. Janzing. On the computational power of molecular heat engines. *J. Stat. Phys.*, 122(3):531–556, 2006.
- [19] L. Schaeffer. A physically universal cellular automaton. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 237–246, New York, NY, USA, 2015. ACM.
- [20] Game of life. Wikipedia. accessed 22 May 2016.
- [21] N. Margolus. Physics-like models of computation. *Physica D: Nonlinear Phenomena*, 10(1):81–95, 1984.
- [22] L. Schaeffer. A physically universal cellular automaton (website). <http://web.mit.edu/lrs/www/physCA/>, accessed February 2016.
- [23] V. Salo and I. Törmä. A one-dimensional physically universal cellular automaton. preprint [arXiv:1501.03988](#).
- [24] C. H. Bennett. Time/space trade-offs for reversible computation. *SIAM J. Computing*, 18(4):766–776, 1989.
- [25] W. Zurek. Algorithmic randomness and physical entropy. *Phys Rev A*, 40(8):4731–4751, 1989.
- [26] D. Janzing, R. Chaves, and B. Schölkopf. Algorithmic independence of initial condition and dynamical law in thermodynamics and causal inference. *New Journal of Physics*, 18(093052):1–13, 2016.
- [27] R. Balian. *From microphysics to macrophysics*. Springer, 1992.